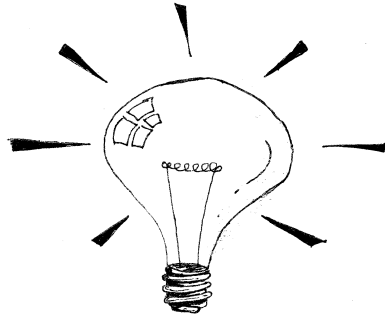


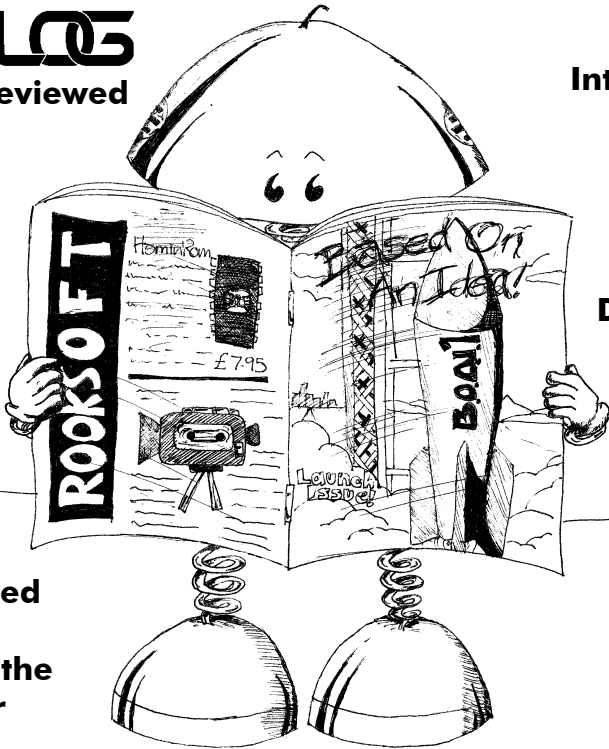
# Based On An Idea...



The Z380 reviewed inside!

How To Program the SAM Mouse

Play Sampled Sound at 10.4kHz! – the BurstPlayer Explained



SAM On The Internet – WWW, FTP and Email Resources

Dalmation BBS – SAM Goes Online!

The SAM Disk Protector Reviewed

• Issue 1 • January 1996 • Price £2.00 •

Launch Issue!

# ROOKSOFT

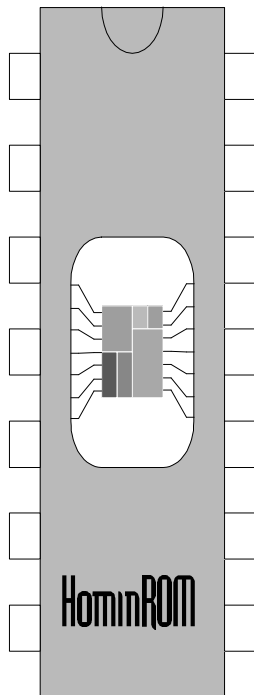
## The HominROM

WORRIED ABOUT THE RISK OF THEFT OF YOUR SAM? Own a pre-version 3.0 SAM ROM? Well ROOKSOFT have a solution to both of these problems...

The HominROM is a v3.0 (equivalent to West Coast v3.1) SAM Coupé ROM, personalised with your own name and postcode on the title screen, making identification much easier in case of theft! (On request we can put any other information in there that you may require – up to a maximum of 32 characters).

This revised ROM also replaces the cassette based function keys with much more useful disk based options such as DIR, BOOT 1, etc.

The HominROM costs just £7.95 and comes with detailed fitting instructions.



## Other Services

### EPROM PROGRAMMING SERVICE

We will program most EPROMs from code files. £1.50 for each EPROM programmed, plus the cost of any EPROMs which we supply; if you prefer, you may supply your own. We will program any quantity.

### PCB DESIGN SERVICE – AS USED FOR THE QUAZAR

PCB artwork produced from circuit diagrams, schematic drawings or netlists. Photo etch-resist acetates, PostScript files or Gerber plot files returned. P.O.A.

### SCHEMATIC DRAWING SERVICE

Circuit diagrams drawn, printed & netlists created from your drawings. P.O.A.

For more information send an S.A.E. to the address below.

1 Dovey Close, Astley, Tyldesley, Manchester, M29 7NP  
Tel: (01942) 886084 • Cheques made payable to M. Rookyard

---

---

# Based On An Idea...

Issue 1 – January 1996

---

---

# CONTENTS

---

## Regulars

- 4 Editorial
- 7 News
- 29 Subscriptions
- 37 Contacts
- 38 Index of Advertisers
- 38 Next Issue
- 38 Credits and Disclaimer

---

## Features

- 14 **How to be a complete Bursterd**  
Stefan Drissen explains how the Burst player routine used in his SAM Mod Player works.
- 22 **Of Mice and Men...**  
The SAM Mouse Interface – in this issue, Simon Cooke documents the software needed to read your rodent.
- 27 **Is there anybody out there?**  
Dave Whitmore's Dalmation BBS goes online! Read all about it!
- 30 **SAM Gets Wired...**  
Where to find the SAM on the World Wide Web, and the rest of the Internet.
- 33 **A High Speed SAM?**  
Martin Rookyard looks at the Z380 with a view towards using it in a SAM accelerator.



---

## Reviews

- 11 SC\_Disk Protector

# Editorial...

Welcome to the first ever issue of *Based On An Idea...* – the brand new technical magazine for the SAM Coupé!

You may be thinking “Why do we need another SAM magazine?” Well, as SAM owners ourselves we’ve read a fair few other magazines in our time and while they’re ok as far as they go, we couldn’t help noticing the relative dearth of good, heavy, technical stuff - so here’s a magazine full of good (we hope), heavy, technical stuff. Admittedly it’s a little sparse at the moment in terms of actual numbers of contributors, but that’s where you lot come in. If you’ve got an article you’d like to write for the magazine, send it in! If it’s good enough we’ll print it (because the mag isn’t called “Based On Our Ideas”) and give you a couple of free issues into the bargain. We want technical articles; hardware projects, the sort of things you haven’t come across in other mags, or things that you have seen but you can do faster or better. What we don’t want are games and things of that ilk - not because we don’t like them, but because other people are already covering that more than adequately. Of course, if you want to do a step-by-step guide to, say, collision detection or fast sprite printing, or even a complete guide to writing a game, that kind of thing is more than welcome!



How the Accelerator works: “Well, you just plug this thing into the back of the SAM, threaten it with a screwdriver, and your SAM’s running at 20MHz before you can blink...”

One of our columns (called “SAMizdat”) will be for forthcoming projects or projects (both hardware and software) under development. Unlike some magazines, we do not operate a policy of only printing something about a project when it is totally finished. We feel that this causes a developer to feel isolated and prevents useful pooling of knowledge. So, if you are working on something, then shout it to the world through the pages of *Based On An Idea...* – who knows, there might be someone out there willing to fund your project, if they hear about it and like it. The flip side of that, of course, is that we can’t guarantee that all of the projects which we tell you about in the SAMizdat section will ever see the light of day. But that’s not the point – at the end of the day, it should help boost the morale of other developers and SAM users at the very least.

You'll also notice a little questionnaire with this issue, please take a few moments to fill it in, to help us put together the kind of mag that you'll enjoy reading (if you don't want to answer any particular question that's ok by us). Send it back to us asap, but before March 1st because on that date we'll pull five winners out of the hat, each of whom will receive the by now almost customary couple of free issues. You can, of course, send it in after then (and please do – we want to know what you think of us!), but you won't have the chance to win ...

Remember, this magazine is meant to be your technical magazine with your ideas, your projects and a place for you to discuss technical matters with other like minded SAM users. If you don't understand something, ask, and there will most probably be someone who knows the answer. In this vein, we'd also like you to write to us with letters which we can include in the magazine. There is going to be a sort of revival of the Spec Tec Jr column which Simon Cooke wrote for Your Sinclair (or something close to it – the short stories, for example, may bow out gracefully in our version), where you can pose any SAM related questions you want or need the answers to. We're also planning to have a Mart section where you can place free adverts for things that you'd like to sell or buy (so send those adverts in now! Keep them of reasonable length though), and the Contacts section is already up and running – but if you're not in there and want an entry, just let us know and we'll put you in.

As for adverts, well, we'd love to be able to give everyone free ones, but we couldn't afford to do that really. As this is a non-profit magazine, we're not going to charge much for them – just enough to (hopefully) cover our printing costs in the long run. If you want to place a full-page, half-page or quarter-page advert in the magazine, get in touch with us and we'll quote you the prices.

All in all though, this magazine is about sharing information. We've got lots of features lined up for the next few issues (most of which are going to be written by Martin and Simon, with the odd guest writer popping up – at least at first, though Simon is dreading what this is going to do to his end-of-degree exams), and we hope that you find this first issue a joy to read, and something to refer back to when you need that specific piece of information that you just couldn't find anywhere else.

These are interesting times for the SAM – we're threatened with obsolescence from all sides, and lots of people laugh at SAM owners for owning ... well, what is to be honest a quirky, underdeveloped little machine. There's surprisingly little software available on a machine which is six years old, and also surprisingly little hardware – but this is beginning to change. Oh, and just to make you nervous – don't forget that the mean time between failures for the components in your SAM, as specified by the manufacturers, is on average five years... and if you bought your SAM in the days of MGT, your SAM is about a year older than that by now!

Simon Cooke, Martin Rookyard, Maria Rookyard, January 1996

## The SAM's Premiere PD Library

# SAM PD

18 Mill Lane, Glenburn Rd, Old Skelmersdale, Lancs, WN8 8RH. Tel: 01695 731163

We have a large selection of Public Domain (and copyrighted) software in stock to suit all tastes. A PD disk catalogue is available – send £1.00, or a formatted disk and an SAE. Also includes some useful progs & a Tetris game! All cheques should be made payable to SAM PD.

### SAM COUPE ARCADE DEVELOPMENT SYSTEM (SCADS) • £2.50

SCADS is now Public Domain. The disk contains the Designer and Supervisor programs, and comes with eight playable demos and a read-me file. The SCADS manual can be obtained from Revelation Software and also from us for £12.95 (200 pages, A4).

### AMSTRAD NC100 NOTEPAD TO SAM • £1.50

Convert Notepad files to SAM format and vice versa. The files created can be used on most SAM wordprocessors. Excellent step-by-step read-me files will take you though all you need to know to transfer files and how to make up the necessary connecting lead.

### PAW CONVERTER • £1.50

The Professional Adventure Writing system can now be used with full SAMDOS compatibility. The original PAW, version A17C, on disk or tape is needed. Contact us if interested in this disk, but don't have A17C.

### DJ0HF AMATEUR RADIO SUITE (Shareware) • Unregistered £1.50 • Registered £5.00

Transmit & receive radio data such as RTTY, CW, SSTV and FAX. The program can control an external TNC. To use, you need a SAM running MASTERDOS with a minimum of 256KB, 1 disk drive and the SAMBUS to provide timing signals. No other external hardware is needed to decode the signal from your amateur radio receiver.

### UTILITY COLLECTION • £1.50

Includes Data Manager - database. Dir-Util - very nice disk manipulation program with pull down menus, handy Diary program, a full sector editor, a stand-alone auto Unerase program, which will restore any erased program without asking you for the file type. SAM-to-BMP converts SAM screens to Windows format graphic files, BMP-to-SAM does the reverse. Text-to-note converts PC text files to Tasword format, note-to-text does the opposite. Fred-to-text will decompress FRED disk magazine text files to normal text.

### 007 DISK DOCTOR • £1.50

Originally sold for £9.95, the disk contains five programs. Disk Doctor, Sector Editor, Close up, Map-it and Exchange. A readme manual file is also included on the disk.

### 007 DISASSEMBLER & 007 REVEAL • £1.50

Three disassemblers for disassembling code at different memory addresses. Reveal is a program for displaying text in code, find high score, names and hidden messages. Also on this disk is an updated disassembler version by Mike Haine, that is also a monitor program.

### SAMART. • £1.50

An excellent little Art Program, which is very user friendly. SAMArt uses two screens, a work screen and an option screen. Also contains a slideshow. Requires MasterDOS and the SAMCo mouse.

### F9 SOFTWARE – NOT PUBLIC DOMAIN!

#### KAPSA EMULATOR • £8.50

A Speccy emulator with lots of extra features. Use it to Snapshot or convert files. K.E.'s ROM lets you use every key of SAM's keyboard and has a help screen of the Speccy's keyboard. Three built-in monitors (Z80-list, Vast and Wlezley - good for viewing graphics) with an MC tracer program for hackers. You also get Ancient Fighters and Tetris III free of charge.

#### ZIPPER (ARCHIVING PROGRAMS) • £8.50

Screen compressor, MC Data compressor and Disc compressor - the excellent Archiving program - archives selected files on average to 50%. FAST decrunching occurs automatically on loading.

#### SAMSPRITE • £9.95

A pointer driven Sprite Designer program, for designing 8x8 or 16x16 sprites. Animation feature and printed manual.

#### EDI-PRO WORD PROCESSOR • £13.50

Full WYSIWYG system, proportional printing, up to 15 fonts in one text file, pull down menu system, integrated font editor, imports text from Tasword and Outwrite, search & replace, screen saver, macros, 32 maths & other symbols, auto-save, and many other functions we've not got space to mention! Edit up to 6 text files at once! Prints to Epson and compatible printers.

#### SAM MOD PLAYER v2.2 • £5.00

The latest version of this excellent MOD player, compatible with SAM Soundchip, EDDAC, Blue Alpha sampler and Quazar Soundcard. Comes with source code to let you use it in your own programs!

Many other Public Domain disks are available from us! Send an SAE to our address for more details.

# NEWS

## **PRODOS Author Convicted**

Chris Pile, author of PRODOS, the SAM Coupé's CP/M2.2 compatible operating system was sentenced late last year (15th November) to 18 months imprisonment under section 3 of the Computer Misuse Act of 1990.

According to a CERT press release, using the pseudonym Black Baron, Christopher Pile, an unemployed 26 year old from Efford, Plymouth created the viruses Pathogen, Queeg and Smeg. At his trial on 26th May 1995, Pile pleaded guilty to eleven charges arising from his creation and release of these viruses. Ten counts related to instances where organisations had suffered unauthorised modification of their computer data by one of these viruses. The eleventh charge relates to inciting others to create computer viruses and hence cause unauthorised modifications.

Although Pile's trial was in May, the sentencing was delayed until November to allow both defence and prosecution counsel to argue the seriousness of these crimes. Pile's viruses were available on computer bulletin boards and on systems connected to the global Internet.

Christopher Pile is the first person in the United Kingdom to be convicted of writing and distributing computer viruses. He is the first person in the

world to be convicted of inciting others to create computer viruses. At the trial in May, Judge Jeremy Griggs described the case as unique and said it was "a dangerous practice to have engaged in".

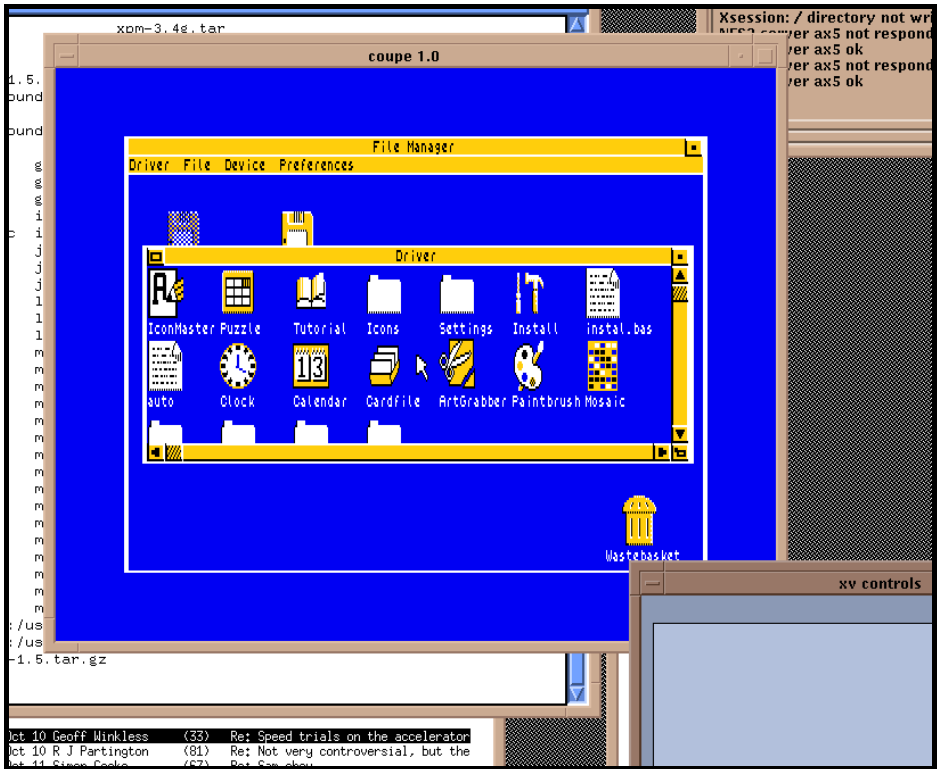
In October 1992 three Cornell University students were each sentenced to several hundred hours community service for creating and disseminating a computer virus.

Unauthorised modification of information in a computer system is an offence under section 3 of the United Kingdom's Computer Misuse Act 1990. The maximum punishment under this section is five years imprisonment or an unlimited fine or both.

## **Is it real, or is it...**

Entropy member Allan Skillman, author of the Driver Mines program which appeared on FRED issue 63, has announced the arrival of a SAM Coupé emulator for UNIX systems running under X Windows.

Having worked on Xcoupe for over a year, the results are startling – and it looks just like the real thing. A number of insights into the design of the SAM were made during the coding of the emulator – including that Darren Clarke, programmer of the Driver Mosaic game, was able to run a routine under interrupts, even though his interrupt handler was testing for the absence of interrupts. (Due to a fluke in the timings, the program still runs).



Driver... running under Xcoupe, under Xwindows, under Unix, under...

We can verify that indeed, it does emulate a SAM rather well – even down to line interrupt levels. It also manages to successfully emulate the VL1772-02 disk controller, so existing DOSes can be used without modification. At the time of writing, there is a problem running most ESI demos – they appear to prove too tricky for it. According to Allan, this should be fixed soon.

You can see the results for yourself by making a visit to Allan Skillman's Xcoupe web pages, at:  
<http://www.hep.ucl.ac.uk/~ajs/xcoupe>

### Sounds bad for the SAM...

Stocks of the SAA1099 chip, used to produce the sound on the basic SAM Coupé are running out. Recently, the last ever general batch of the Phillips chips were manufactured, and Phillips will now only be considering minimum orders of 10,000 or more units, to cover the manufacturing and set-up costs.

According to Bob Brenchley of FORMAT Publications, West Coast Computers have bought the last of the stocks in this country, and when Siemens approached Phillips to buy the chip, they were passed on to West Coast.



Siemens are using the sound-chips in railway signalling boxes in order to provide different audible alarms for every possible type of emergency condition.

## High Capacity Storage

The last Gloucester show (October 1995) saw Nev Young of S.D. Software displaying his wares next to the FORMAT stand. While the show seemed a bit on the quiet side in comparison to previous ones, there were three major attractions there – The Quazar, our Accelerator prototype, and the first ever released SAM IDE Hard Drive Interface.

The interface wasn't on sale at the time, but Nev was actively demonstrating its capabilities with a rolling (and seemingly never-ending slideshow). By the time of the Edinburgh SAM Show, the interfaces were ready for sale, and you can now buy the interface, with 84Mb hard drive, PSU with cable, Connecting IDE cable and a provisional DOS from FRED Publishing for £150, which includes postage. For £180, you can buy the same package but with a 120Mb drive, and for £220, the package with a 170Mb drive.

Though we remain sceptical about the efficiency of the filing methods used on the drive, it should work with any IDE drive available on the market (including the recently released EIDE drives, which correspond to the ATA2 specification).

When we asked him, Nev Young explained that the file directory is held

as one large file in the system. Subdirectories are marked in a similar way to MasterDOS subdirectories, but use a 16-bit word to store the directory tag – enabling the user to have up to 65535 subdirectories on the hard drive.

We think that there may be speed problems associated with this, especially when the number of files on the hard-drive becomes very large. But we're remaining optimistic – some kind of hard drive is better than none, and no doubt if the DOS isn't up to scratch, someone will release a new one in time.

If you would like to buy one of the interface packages, contact FRED at: FRED Publishing, 40 Roundyhill, Monifieth, Dundee, DD5 4RZ. Cheques made payable to FRED Publishing.

## Nice & Compact

Speaking of IDE drives, Johnathan Taylor (a PRODOS hacker and Entropy member, currently working on a SAM version of CP/M v3.0) has recently reported that he can now read ATAPI CD ROMs (the kind which plug into an IDE drive socket on the PC) on his SAM, using a Generic Z80 IDE interface. All he needs now is a real CD-ROM disk with a complex directory structure to develop a fully-fledged reader utility. Currently, Johnathan is using two SEGA Mega-CD games to test the system, and it all seems to work fine – though there is a distinct lack of files around which to test the system. More news on this as we have it.

## All Formats Show Dates

Here's the dates for the about-to-occur All Formats Fairs... usually Derek Morgan makes it to the Haydock one (which is also the one that we tend to go to). As for the others, we couldn't possibly comment – which is a face-saving way of saying that we haven't got a clue about whether or not there is a SAM presence.

Jan 20 (Midlands)

NAC (Royal Showground)  
Stoneleigh.

Jan 21 (London)

Tolworth Recreation Centre, A3,  
Surbiton.

Jan 27 (North East)

Northumbria Centre, Washington,  
Dist 12.

Jan 28 (Scotland)

Mitchell Theatre, Charing Cross,  
Glasgow.

Feb 3 (North West)

Haydock Racecourse, Haydock.

Entry is £4 for adults, £2 for children.  
You can order a stand by giving Bruce  
Everiss a ring on either 0181 856  
8478, or 0973 175 131.

## The Edinburgh SAM Show

The recent SAM show in Edinburgh on the 12th of November was quite good fun. Unfortunately, there weren't that many SAM users there, so it tends to pale in comparison with the Gloucester shows, though the fact that it was piggybacked onto another computer fair made it quite a good day out for bargain hunters.

People who made it to the show included Colin Piggot (with the

Quazar Surround – blaring out at full volume across the SAM corner of the show), Allan Clarkson (selling copies of Crashed), Jupiter Software and their games, FRED Publishing, and Bob Brenchley and Nev Young of FORMAT and S.D. Software respectively were also present, so we had a good natter. If you weren't there, why not? :-)

## Quazar Dropped!

Colin Piggot has recouped the development costs of the Quazar Surround 16-bit and 8-bit sound board, and is now able to offer them at a reduced price of £53.99. See his advert in this issue for more information.

## Termite Name Change

Due to possible problems with another piece of software already available for various machines, Simon Cooke's work in progress, Termite, has had to undergo a name change. TERMITE is already a registered trademark for a terminal emulation suite, so to avoid possible problems, Simon has decided to change the name to something else. Current ideas include Photon Comms or just Photon. If you have any better ideas, send them in to the Based On An Idea address!

Email your news to us at:  
**entropy@jumper.mcc.ac.uk**  
or post it to us at:  
**Based On An Idea, 1 Dovey  
Close, Astley, Tyldesley,  
Manchester, M29 7NP**

# Reviews

Only one review this issue – but hopefully quite an in-depth one. Next ish we aim to review at least two things! (Oh Ghod... Ed)

**SC\_Disk Protector, £15.00**  
**Steve's Software, 7 Narrow**  
**Close, Histon, Cambridge, CB4**  
**4XX. Tel: (01223) 235150 from**  
**6pm-9pm, Mon-Sat.**

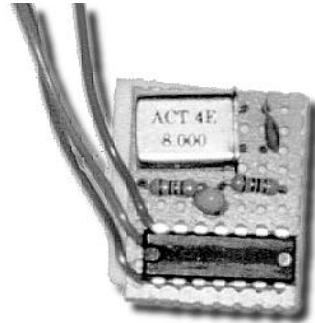
You know the kind of thing. It's probably happened to you hundreds of times before. You're working on your latest masterpiece, your current *raison d'être*, and it crashes. No problem, you think. I've just saved a backup copy of it before running the assembled code (Everyone does this... don't they? If not, it's a very good habit to get into, and can save all kinds of headaches). So you reset the machine, stick your work-disk in the drive, and press the F9 key...

...at which point, of course, the disk just sits there spinning, doing nothing. Perhaps the Missing Disk error comes up. Perhaps not.

Perhaps, as has happened to me with alarming frequency while I've been working on important bits of programming, the disk drive gives a rather ominous Clunk! In which case, as you've probably guessed, large chunks of the disk have just been scrawled over by an errant disk head, thus leading to pain, anger, nausea and the entire drive assembly

pinwheeling its way across the room to embed itself in a nearby bookshelf.

Therefore a welcome addition to the fold is the SC\_Disk Protector, which promises to remove all such causes of woe in a generally painless fashion.



The SC\_Disk Protector: Small but perfectly formed.

What causes the common-or-garden drive hang problem is the fact that the VL1772-02 disk controller chip in the SAM derives its 8MHz clock from the ASIC – as does the sound chip. This would be all well and good, but for the fact that while the RESET line is low, the ASIC stops generating the 8MHz clock signal. Unfortunately, while the sound chip appears to cope quite admirably with this, the disk controller doesn't. (Well, the sound chip doesn't cope either, really. It's just that you don't notice that it's not coping until you try writing your own

ROM routine which turns the sound chip off immediately after a reset, in the hope of getting rid of that annoying flat tone which you often hear if you reset the machine while it's in the middle of playing some music. The sound chip won't respond until at least a couple of hundred microseconds after the reset.)

The VL1772-02 controller chip requires clock pulses while its RESET line is held low to initialise its internal circuitry. Because it doesn't get these, all kinds of weird and unpredictable effects can occur. Nine times out of ten, it's okay, but it's that tenth time which is the damaging one.

Edwin Blink was the person who first discovered this, and he even mentions it way back in his Def Leppard sample demo. The solution is to provide a new 8MHz clock signal to the disk controllers (and also to the sound chip), and the SC\_Disk Protector unit is the board which Edwin designed to do this.

The SC\_Disk Protector arrived here in a large envelope, which contained the protector wrapped in bubble-wrap, and a sheaf of A4 printed instructions on how to fit the device. The protector board itself (see picture above) is quite small – about an inch or so square – and is a piece of veroboard with a square of sticky-backed foam on the back to fix it inside the SAM.

It does require some soldering and some track cutting to fit this into your SAM, so you need to kit yourself out with either a Stanley knife, a scalpel

or a craft knife, a soldering iron and some solder. Alternatively, you can get a local electrician or electronics expert to help you out with the modification to the circuit board.

While the instructions are slightly unclear at some points, there is a diagram included which explains perfectly what goes where. The only thing I could wish for is that the diagram were in colour so that it'd be easier to see which wire was which, as there are three – one red, one blue and one green.

After you've done all the soldering and fitting (which does take a while – I'd advise you to take your time over it, but it shouldn't take much longer than twenty minutes to complete), the board is affixed to the top of the ASIC by peeling off the back of the sticky-backed foam, and pressing it firmly home. This ensures that it won't rattle about – it fits in quite neatly, and there's no stray wires to worry about.

You only need to buy one of these to fix both of your drives, and we can heartily recommend getting one!

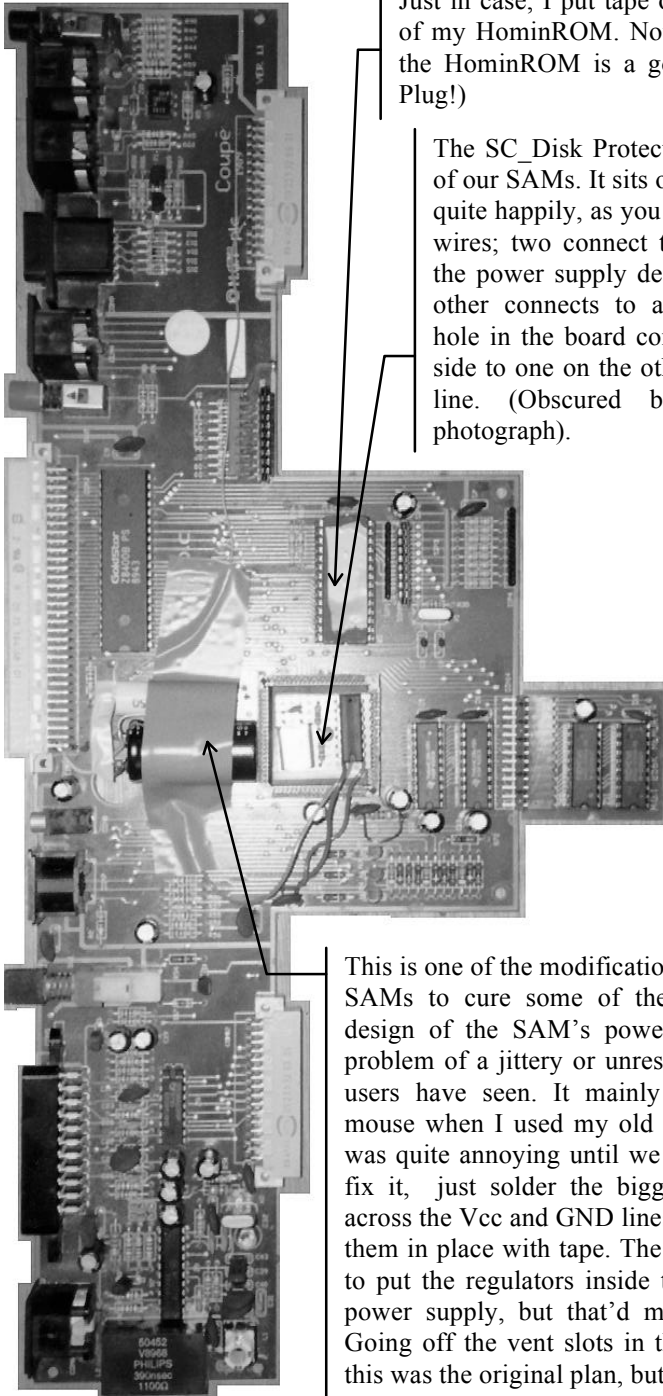
#### **RATING:**

☆☆☆☆☆

#### **SUMMARY:**

**Buy one of these and Reset your machine in complete confidence!**

*Simon Cooke*



Just in case, I put tape over the quartz window of my HominROM. Not strictly necessary, but the HominROM is a good investment! (Plug! Plug!)

The SC\_Disk Protector unit in place on one of our SAMs. It sits on top of the SAM ASIC quite happily, as you can see. There are three wires; two connect to either side of one of the power supply decoupling capacitors, the other connects to a via (an electro-plated hole in the board connecting a track on one side to one on the other) on the 8MHz clock line. (Obscured by the tape in this photograph).

This is one of the modifications which we've made to our SAMs to cure some of the problems inherent in the design of the SAM's power supply. It also cures the problem of a jittery or unresponsive mouse which some users have seen. It mainly happened to my (Simon) mouse when I used my old TV with my SAM – and it was quite annoying until we put the modification in. To fix it, just solder the biggest capacitor you can find across the Vcc and GND lines. We've used two and fixed them in place with tape. The only other way to cure it is to put the regulators inside the SAM rather than in the power supply, but that'd mean redesigning the board. Going off the vent slots in the case, it would seem that this was the original plan, but it never got done. Oh well.

# HOW TO BE A COMPLETE BURSTERD

(or how the Burst Player works - sort of)

In this article I'm going to try to explain how the Burst Player works. Just in case you haven't got a clue what the Burst Player is, here's a short history.

The Burst Player is a very fast routine that Edwin Blink wrote which allows four different samples to be played via the sound-chip at 7.8kHz under line interrupts – thus allowing a sequencer to be written very easily. This opened up the possibility of having a SAM Mod Player. With a minor adjustment, the samples could also be played back at various volume levels. It was at this point that I wrote my first Mod Player, which is nothing more than a sample sequencer. Edwin was not too happy with the sample quality from the soundchip at 7.8kHz so he designed the do-it-yourself EDDAC (now commercially available as the SAMdac). Due to the extra resolution at which samples could be played back (7 bits per sample as opposed to 4 bits) the mods played back sounded even better. Edwin could still see room for improvement, and so he bumped the play-back frequency up to 10.4kHz. Having done that, thinking he had done enough, he left for Thailand. The rest was up to me.

The early Mod Players (on FRED 41 and PRIME 7) used Edwin's code for the soundchip; the EDDAC player (on FRED 52) also used Edwin's code, but now for the EDDAC/SAMdac. The commercial SAM MOD Player uses my own code, and it is the workings of this latest version of the Burst Player that I will try to explain.

## **What's the frequency Kenneth?**

When playing samples it is nice to be able to play them at different speeds so that you can make music, and there are two ways of playing a sample at a different frequency than its original one. The first method is to change the amount of time between outputting two consecutive sample bytes. By increasing the amount of time between outputting new sample data, you decrease the output frequency, and by decreasing the amount of time between outputs you increase the output frequency. You have to output all of the bytes of the sample this way.

The other method is to output sample bytes at fixed intervals. To alter the frequency of the sample being played you simply skip sample bytes or play the same byte twice. This method would, it seems, only allow samples to be played back at multiples of twice the original speed, rendering the

method useless. You can, however, add 1.5 bytes to the pointer each time. An example should help to clear things up.

### Method 1

```

loop:   ld hl,&8000 ;set sample pointer
        ld a,(hl) ;get sample byte
        out (&E8),a ;output sample byte
wait:   ld b,&30    ;wait loop
        djnz wait
        inc hl    ;increment sample
                ;pointer
        jr loop

```

### Method 2

```

        ld hl,&8000 ;set sample pointer
                ;(integer part)
        ld e,&00    ;set sample pointer
                ;(fraction part)
        ld bc,&0180 ;set increment (b is
                ;integer, c is
                ;fraction)
loop:   ld a,(hl)  ;get sample byte
        out (&E8),a ;output sample byte
        ld a,e    ;1]
        add a,c   ;1] update fraction
                ;part of sample pointer
        ld e,a    ;1]
        ld a,b    ;2]
        adc a,l   ;2]
        ld l,a    ;2] update integer
                ;part of sample pointer
        jr nc,$+3 ;2]
        inc h     ;2]
        jr loop

```

As you can see in the method 2 example above, the sample pointer is now defined by three bytes - two for the integer part (register pair hl) and one for the fraction part (register e). The sample pointer can now theoretically point to address

32768.25 (hl=32768, e=0.25\*256=64), the content of this address is of course identical to the content of the integer part. So now if we want to play a sample at 1.5 times the original speed then we simply set b to 1 (integer part of 1.5 = 1) and c to 128 (fraction part = 0.5\*256). The main advantage of this routine is that we don't have to set up a dummy loop which takes up all the processor time but can go and do something more useful.

Now all we need is something which will ensure that our output sample byte routine is called at a fixed time interval. Unfortunately the SAM Coupé does not have a programmable timer interrupt so that we can tell the processor to generate say 10400 interrupts per second. What we do have is the line interrupt. The only problem with the line interrupt is that you can only get one on the screen lines (0 to 191) and not during border time. Before going into this problem I'd like to continue on the more general theme and cover different volumes.

## IS THIS LOUD ENOUGH?

If you want to use samples to create music then you will want to be able to change the volume of the samples. How else would you be able to create echoes or add depth to the piece? One very basic solution would be to have a number of samples all recorded at different volume levels. This would take up large amounts of memory. Going along and calculating everything real-time would be very

time consuming. Consider the following partial example.

```
ld a,(hl) ;get sample byte
ld b,&80 ;multiplier
;(&00=0%, &FF=100%)
call multiply ;call multiply
;routine to get required volume
out (&E8),a ;output sample byte
;(with new volume)
```

Since multiplying is not one of the Z80's strongest points we will need to get around this in another way. The usual way to get round calculating stuff real-time is to put it all into a table. Since an 8-bit sample byte can have 256 ( $2^8$ ) different values we'll need 256-byte tables. In the Burst Player there are 32 volume tables, each starting at a 256 byte boundary. Now the routine looks something like:

```
ld d,volume ;d is high byte of
;volume table
loop: ld e,(hl) ;get sample byte
ld a,(de) ;a is now a sample
;byte with volume
out (&E8),a ;output sample byte
;(with new volume)
```

The contents of the volume table is dependent on the sound device. If we need to output at 7 bits per channel we will need the bytes in the volume table to be in the range [0..127]. For a 4 bit per channel device (the soundchip) the bytes in the volume table will be in the range [0..15]. With the soundchip we actually have another problem which is easily solved by working with volume

tables. The soundchip volumes are in stereo. The highest four nibbles are right, the lowest four nibbles are left. Normally we would need to do the following:

```
ld d,volume ;d is high byte of
;volume table
loop: ld e,(hl) ;get sample byte
ld a,(de) ;a is now a sample
;byte with volume
rlca ;]
rlca ;] shift bits for
;stereo right
rlca ;]
rlca ;]
out (&E8),a ;output sample byte
;(with new volume)
```

However, we can incorporate these shifts into the volume table. The minor disadvantage of this is that we can only have half the amount of volume tables in the same amount of memory since we will need two volume tables for each volume level. One table for left stereo and one for right stereo.

Now that we know what the principles are behind outputting a sample at a variable frequency and volume we can see how we are going to get four samples pumped out at once.

### Two's company, three's a crowd, four's a hassle

Firstly, if you want to output four samples at once on the SAM the biggest problem is how to do this as quickly as possible. Fetching the samples from memory real-time will not work unless you make sure that you fit all your samples into two



pages. By doing this you can have your samples paged in all the time – however, 32k is not exactly a massive amount of memory for samples. We want more! Using more memory will mean that you will have to page the sample data into the memory-map every time you need to read a sample byte. If you are fetching sample bytes real-time you'll be spending more time paging the data in and out than anything else. The whole idea behind the Burst Player is that it uses two buffers of 208 bytes to store the samples per frame. Why 208 bytes? Well, 208 bytes per frame equals 10400 (50\*208) bytes per second = 10.4khz.

The two buffers are used alternately. While one buffer is being filled with data, the data from the other buffer is sent out to the sound device. Once that buffer is empty, the buffers are swapped round, so that the one that is now full is used to produce the sound, and the other is filled up with data. What advantage does this have you may ask. By working in this way you only have to deal with the sample data for one channel at a time, rather than all four channels at once. Therefore you only need to page in the right page of the sample, transfer the bit you need to the buffer and then do the next channel until you reach channel four. Another advantage is that you only have to set your sample pointer, speed and volume at the start of each 208-byte fetch routine.

### **What time is love?**

Now that the basics of the Burst Player have been covered we can

move on to that tricky problem mentioned earlier, timing. Timing is a wobbly thing on the SAM. This is due to memory contention when the screen is on. For this reason just about every early sample demo turned the screen off so that there would be no contention, and thus they would not have to worry about their samples sounding as if they were underwater. This was until Edwin came up with the line interrupt sample player (the notorious Kim Wilde and Def Leppard demos). The whole idea behind the line interrupt sample players is that they use the line interrupt to generate the required timing pulses without having to worry about memory contention.

As I mentioned earlier, line-interrupts are only generated in the graphics area of the screen display. While the ASIC is displaying this area of the screen, the SAM's memory is contended and the timing consequently goes all wobbly. Fortunately, while the graphics area is being displayed we can use the interrupts to ensure that our sample timing remains accurate. We can't generate interrupts while the ASIC is displaying the border, but during this time the SAM trots along at an even pace so we can count on the processor to keep time.

The Burst Player therefore starts in the border (at the bottom of the screen) using software timing, as soon as it is at the top of the screen, entering the screen area, control is passed over to line interrupts to finish the job.

## Cocktail

Since the Burst Player is constantly busy, the most logical place to begin explaining what it does is at the beginning of the border area below the screen area. The routine needs to fill one of the two buffers with sample data that is in a suitable form to be directly output to the sound device that the user has selected. For all devices except the Quazar this requires mixing. One byte contains 8 bits, and therefore can contain one 8-bit sample. You can fit two 7-bit samples into a single byte by simply adding them up (the eighth bit being carried from the addition of the 7-bit numbers) – this is called mixing. When the Burst Player needs to mix two channels together it does the following:

```
chan.one:...  
    ld e,(hl)    ;get sample byte  
    ld a,(de)   ;get sample byte  
    ;including volume  
    ld (mix.4+1),a    ;put sample  
    ;byte in channel four routine  
    ...
```

The sample data is inserted directly into the routine of the channel with which it needs to be mixed – this self-modifying code approach being quicker than storing the value in a separate memory location. The routine for channel four then looks something like this:

```
chan.four:...  
    ld e,(hl)    ;get sample byte  
    ld a,(de)   ;get sample byte  
    ;including volume
```

```
mix.4:  add a,0      ;mix channel four  
        ;byte with channel one byte  
        ld (buffer.0),a ;put result in  
        ...         ;buffer
```

This method is very fast. It has one disadvantage though, in that all the addresses have to be put into the code resulting in the largish (32k) size of the Burst Player.

The Quazar does not require any mixing to be performed because it has six 8-bit sample channels – and most mods only require four channels. However, it does need buffers which are twice the size of the buffers needed by the other devices to store the extra channel data – instead of putting the result of channel one into the routine for channel four it puts the result directly into the proper buffer.

## Repeat, repeat, repeat, repe...

To return to the code starting at the bottom of the border, the same instructions repeat (with different addresses) to fill the buffer. After a given number of instructions a “dump sample to device” routine is inserted into the code so that the output frequency remains as close to 10.4 kHz as possible. This is a form of software timing, but since the code is started by an interrupt and uses software timing during the border area there is one small problem. When an interrupt is generated the Z80 finishes the current instruction cycle and then jumps to the interrupt routine. The problem is that if the current instruction was a NOP then the Z80 would respond to the interrupt 4 t-states after it occurred, however if it was an LDI then the Z80 would not be

able to respond for 16 t-states. This 12 t-state difference can put the timing out of sync, leading to the samples sounding as if they were underwater. To fix this little problem the last used line interrupt does not pass control back to the main code but enables interrupts and then does a HALT, thus ensuring that the line interrupt which starts off the border routine always occurs at exactly the same position on the screen for every frame.

Once the TV scan line has passed the border area at the top of the screen the buffer is filled and control can be passed back to the main program. Line interrupts are set up to play back the rest of the first buffer through the screen area. This goes on until the TV scan line reaches the line above the bottom border (line 191) and then the buffers are switched over and the whole process repeats.

## Humpty Dumpty

It's all very well talking about timing, line interrupts, volume tables etc., but what about actually outputting the sample data itself? The Burst Player can dump samples to six different devices (Colour Look Up Table [CLUT], Soundchip, SAMdac, DAC, Blue Alpha Sampler and Quazar Surround) and therefore needs six different output routines. Sure, the CLUT is not exactly your ideal sound device (for one thing, you can't actually hear what you're playing through it) but it is useful in that it gives you an idea of what is going on in the output stage. For all of the output routines the alternate register pairs BC' and DE' are set to certain

values and HL' points to the position in the buffer. Each dump to an output device does not affect the contents of the constant registers.

First up on the catwalk is the rather silent CLUT routine:

```
;BC = 248 (CLUT port, pen 0)
;DE = &1734 (the two colours for the
stripes, blue and purple)

sd.clut:
    out (c),e    ; set first control stripe
                ;(purple)
    inc b        ; an OUTI first does a
                ;DEC B
    outi        ; (DEC B) send out
                ;data
    out (c),d    ; set second control
                ;stripe (blue)
    inc b        ; since OUTI first
                ;DECs B
    outi        ; (DEC B) send out
                ;data
```

Now on to the bit which all SAM users can use (except for those with knackered SAA1099's – I managed to do this to mine with a faulty scart cable):

```
;BC = &1FF (soundchip register select,
;255 = soundchip data)
;DE = &0205 (&02 = volume channel
two,
;&05 = volume channel five)
;to initialise soundchip for sample
;playback:
;- clear all registers
;- turn on the chip (28,1)
;- set envelope generators 1 and 2 to a
;sustained waveform (24,130; 25,130)
;- be polite and set volume 2 and 5 to the
;silence volume level (usually
;:%10001000)
```

```
sd.saa:
    out (c),e ; select channel five
    outi      ; (dec b) output data
    inc b     ; reselect register control
    out (c),d ; select channel two
    outi      ; (dec b) output data
    inc b     ; reselect register control
```

For the oh so sexy SAMdac (available for £25 – nudge nudge, wink wink) the following routine comes rolling in:

```
;BC=232 (printer port 1) or 234 (printer
port 2)
;DE=&0001 (strobe selects)

sd.samdac:
    outi      ;(dec b) output
                ;sample byte
    inc c     ;select strobe port
    out (c),e ;select channel 1
    dec c     ;select data port
    outi      ;(dec b) output
                ;sample byte
    inc c     ;select strobe port
    out (c),d ;select channel 0
    dec c     ;select data port
```

For the people who soldered together a basic printer DAC the following routine is used:

```
;BC=232 (printer port 1) or 234 (printer
;port 2)
sd.dac:
    ld e,a    ;store the A register
                ;(EXX is used for interrupt)
    ld a,(hl) ;get sample byte 1 in A
    inc hl
    add (hl) ;mix sample byte 2 into
                ;A
    inc hl
    out (c),a ;output sample byte
    ld a,e    ;restore the A register
```

The routine for the Blue Alpha Sampler is identical to the printer DAC routine except that the Blue Alpha Sampler needs to be initialised to play-back the samples and the output port is different:

```
;BC=&7CFE (data port of Blue Alpha
;Sampler). To initialise the Blue Alpha
;Sampler for sample playback:
;- OUT &FEFE,&FF
;- OUT &7DFE,&FD

sd.bas: ;identical to sd.dac
```

Finally, the Quazar Surround Soundcard uses the following routine:

```
;BC=&06D0 (output port)
;DE=&0006 (store to reset B)
;To initialise the Quazar, IN &06D0 to set
;to mode 1 (6x8-bit mode)

sd.qss:
    ld b,e    ;reset the B register for the
                ;output port
    outi      ;(dec b) output sample
                ;channel 3
    outi      ;(dec b) output sample
                ;channel 2
    outi      ;(dec b) output sample
                ;channel 1
    outi      ;(dec b) output sample
                ;channel 0
```

Phew, that covers that. If you look at the different output routines carefully then you should notice something odd about the Quazar output routine. The Quazar outputs four bytes from the buffer each time it is used, whereas the other routines only output (or use) two bytes. The reason for this is that

the Quazar is the only output device which allows you to output 8-bit samples. All of the other devices can output a maximum of seven bits per channel (even though some devices have 8-bit outputs, we still need to combine some channels together so that you can hear them all!). By mixing two seven bit samples together you can store the sample data in half the space that the Quazar needs to store all four of its samples.

### Encore

That covers that. I hope you now have a vague idea of how the Burst Player works. If you didn't understand any of it, don't worry as it's not exactly the easiest piece of code in the world to follow. For those of you who did understand it, and who are actually slightly 'into' sample routines themselves, how about incorporating support for the other sound devices as well (and not just the soundchip)?

Now how about somebody writing some more cool sample demos, based on an idea read about in.....?

*Stefan Drissen*

# Available NOW!

Limited stocks so don't delay!\*

## SAM MOD player version 2.04

- Output frequency of 10.4kHz (with screen on!)
- play Amiga MODs on your SAM via the soundchip, SAMdac, printer dac, Blue Alpha Sampler or Quazar.
- ALL Protracker commands implemented so you can hear your MODs how they were meant to be heard! (even BPM changes and the Extended effects are implemented)
- Load MODs directly from PC or SAM discs (the file selector also only shows MOD files so you don't need to wade through the directory looking for that MOD file - handy for FRED discs!).
- COMET source file showing you how to use MODs in your own code!

All this for only FIVE pounds cash (or Eurocheque for 12.50 dutch guilders). For an extra £5 (etc) you also get the FULL COMET SOURCE!!!

## SAMdac

- play MODs at 7 bits per channel in stereo! (8 times the soundchip's quality - half the quality of the Quazar).
- connects to your printer interface and audio out port (interface has a built-in through port).
- FREE copy of the latest version of the SAM MOD player!

Sick of listening to your SAM's feeble attempts at reproducing Amiga MODs but not prepared to shell out for the Quazar? Then the SAMdac is the answer for you, priced at only £25 cash (or a Eurocheque for 65 dutch guilders).

BOTH products are available from:

Stefan Drissen, Zevende Herven 6, 5232 JZ 's-Hertogenbosch, The Netherlands.  
Eurocheques payable to Stefan Drissen.

\* fortunately stocks are not limited - it sounds spectacular though doesn't it?

# Of Mice and Men...<sup>o.o.o</sup>

## The SAM Mouse Interface Explained

It's unfortunate really. The SAMCo mouse came out near the beginning of 1991, yet hardly any software actually uses it. Like a good proportion of the SAM hardware that has come along, programmer's documentation has been thin on the ground. This has led to a few problems.

Most SAM software today doesn't use the mouse – even though in some cases, it would be ideal for the job at hand. It's a shame – some people would at least like the option. So, in our (decidedly less than) infinite wisdom, we've reverse engineered the mouse interface itself, and properly documented this wee beastie... well, at least we hope we have.

We're going to take it from a software point of view first, and then next issue we'll get into the nitty-gritty and explain how the hardware works. We might even come up with a circuit that enables you to use a PC serial mouse in the SAM mouse socket. This will probably involve the use of a PIC controller, which should cut down component count considerably. I've already had a PC mouse working on my SAM through the Comms interface (albeit a little jerkily due to the basic "see if it works" pointer routine I was using), so even if we don't get it working that way, there are still other methods available to us.

### Archaic Artefacts

The mouse is read in a similar fashion to the keyboard, using port &FFFE.

In the technical manual, this is referred to as the RDMSEL port. The only other mention the manual makes of the mouse is that bit 1 of the ASIC interrupt status register refers to the mouse interrupt. This is an error, left in the manual from a very early stage of the SAM's design. That bit of the status port should actually refer to an external or Comms interrupt, and I've been using it extensively in my own terminal software. To do so requires a small mod to the board – no problem!

### The meaning of life

You may be wondering why the mouse interface is so big. This is because its purpose in life is to decode pulses from an Atari ST style mouse, and to translate them into a form that can be easily read from the SAM's mouse port. Originally, the plan was for MGT to produce their own mouse, with a tiny ASIC inside. Alas, this didn't happen, so we're stuck with the big box.

All you need to know at this stage is that the interface keeps track of how far the mouse has moved since you last read the data from it. Next issue

	3	2	1	0
Dummy data	1	1	1	1
Button status	/BUT4	/BUT3	/BUT2	/BUT1
Y256	Y800	Y400	Y200	Y100
Y16	Y080	Y040	Y020	Y010
Y1	Y008	Y004	Y002	Y001
X256	X800	X400	X200	X100
X16	X080	X040	X020	X010
X1	X008	X004	X002	X001
Dummy data	1	1	1	1

Table 1: SAM mouse data format.

we'll actually go into the mechanics of the interface.

### Talk to the animals

This is where you get to be your own personal version of Dr. Dolittle. When the mouse wants to transfer data to the SAM (when you read RDMSEL, for example), it puts the information on the lower four bits of the bus. If you want to read the data back, this will mean a little bit of rotating, but never mind – that's the easy part.

The output format of the mouse data can be seen in Table 1. It's quite simple really. To read the mouse, the first thing you must do is strobe it – all this involves is reading the port once. Then you can start actually bringing in the mouse data (but don't forget to mask it with &0F to get rid of the upper bits!).

The first value that you get is a dummy one. This apparently useless data is actually quite important – it allows you to check if the cursor keys have been pressed. If they are, you won't get a value of &0F for the dummy byte, and you can delay

reading the mouse until they have been released. This ensures that the data you read won't be garbled.

The next thing to do is to read the mouse port eight times, and store the data in an 8 byte long buffer. You cannot take any longer than ~30µs between each read of the mouse, otherwise the mouse will reset its internal counters, and you'll be back where you started. If you use something along the lines of the code below, you should have no problems with timing.

#### Mouse Reading Routine\* :

```
rd.mouse:
    LD BC,&FFFE
    LD HL,mouse.data
    LD DE,&060F
    IN A,(C) ; strobe mouse
    IN A,(C) ; read first data byte
    AND E ; mask off upper bits
    CP E ; check if it's equal to
           ; &0F
    RET NZ ; if not, we abort
```

\* This routine is loosely based on the mouse driver provided with the mouse itself, which was written by Dr. Andy Wright.

rdloop:

```
LD (HL),A ; store in data area
INC HL
IN A,(C) ; read a new byte
AND E ; mask it off
DEC D ; have we finished?
JR NZ,rdloop ; if not, loop.
```

```
LD (HL),A ; store last data byte
```

The more astute of you will have seen that there are some easy ways of optimising this routine — for example, if the 8 byte data area which HL points to is inside a 256-byte page (ie you can write to every byte in the buffer by changing only the bottom 8 bits of its address – in other words, by changing the L register), you can use INC L instead of INC HL.

If you look at Table 1, you could also discard the last byte of data read from the mouse, as it's a dummy byte. It's not really supposed to be a data byte at all – the final read is actually necessary to reset all of the mouse interface's internal counters to zero. Therefore if you wanted to, you could reduce the data buffer to 7 bytes, and get rid of that final LD (HL),A.

Another optimisation could be made by not storing the initial dummy byte, though that might expand the routine a little – it would still give you an extra byte free in the systems variable area if you needed one. All of these speed-ups are left as an exercise for the reader.

### Where is it then?

The next thing to do is to actually decode the data. How you've read in

the data, and which optimisations you've made will decide how you have to decode some of the data. The button data doesn't need decoding, and you can even read it directly from the buffer area to save space (it'll hold valid data as long as the mouse is plugged in). If a bit is reset, then that button on the mouse has been depressed. On a two button mouse, the left button corresponds to bit 0, the right button corresponds to bit 2. On a three button mouse, bit 1 corresponds to the middle one – but these are much rarer.

Most mouse routines ignore the upper X and Y nibbles – a pity, because if Bruce had known this from the start he could have removed a couple of counter chips from the board and made the interface that little bit cheaper. There's nothing to stop you from using them though – and they should provide a little more leeway if you use the mouse for controlling a game. In this article though, for simplicity we'll ignore them.

### Decoding the X and Y data:

```
maxy: EQU 191 ; this is the lowest
;we allow the mouse to go
```

```
decodexy:LD HL,mouse.data+3
; HL now points to the Y16
; data byte.
LD A,(HL) ; A = Y16
INC HL
LD E,(HL) ; E = Y1
ADD A,A
ADD A,A
ADD A,A
ADD A,A ; A = A*16
OR E ; combine with Y1
LD E,A
```



```
LD A,(Ycoord)
; get current Y coordinate
; (between 0 and 191)
BIT 7,E ; see if we're
; subtracting.
JR NZ,msysub ; jump if we are.
```

If we've got to this point, we're adding a value between 0 and 127 to the Y coordinate.

```
msyadd: ADD A,E ; add offset to current
; Y coordinate.
JR C,hit.bottom
; if Y has overflowed, fix it

CP maxy+1
JR C,yokay
; jump if we've got a valid Y.
```

If we've made it to this point, the value we've worked out for our Y coordinate has either overflowed (ie it would have been greater than 255), or it's greater than the value we've given maxy, so we need to give our Y coordinate a valid value. In this case, we use maxy, so that if it reaches the bottom of the screen, the pointer will stop as if it has hit a wall.

```
hit.bottom:
LD A,maxy
JR yokay

; If we're here, we're subtracting our
; offset from the Y coordinate.
msysub: ADD A,E
JR NC,yokay ; no underflow
XOR A ; set A to zero - top
; of the screen.

yokay: LD (Ycoord),A ; Store our
; new Y coordinate.
```

```
INC HL
INC HL ; skip the X256 byte
LD A,(HL) ; A = X16
INC HL
LD E,(HL) ; E = X1
ADD A,A
ADD A,A
ADD A,A
ADD A,A
OR E
LD E,A ; E holds X offset.
LD A,(Xcoord) ; get our
; current X coordinate
```

```
BIT 7,E
JR NZ,msxsub ; again, check
; whether we're subtracting or
; not.
```

```
msxadd: ADD A,E ; add offset
JR NC,xokay ; X hasn't
; overflowed.
```

```
LD A,&FF
; set X to the highest possible
; coordinate if we hit the edge
JR xokay ; set X coordinate.
```

```
msxsub: ADD A,E ; add offset
JR C,xokay
```

```
XOR A ; X underflowed
; so we fix it...
```

```
xokay: LD (Xcoord),A
```

As you'll have noticed with the above routine, you need two bytes to store the coordinates in – Xcoord and Ycoord. You should set these to zero (or any other initial coordinate you desire) before you read the mouse for the first time. Other things to note are that the range of the X coordinate is between 0 and 255, and that the range of the Y coordinate is between 0 and 191, with 191 being the bottom of the screen, and Y being the top (ie it's

upside-down compared to BASIC – or vice versa, as most programmers would argue).

### Build a better mousetrap...

Have a hack at the ROM source, or alternatively, buy it from Dr. Andy Wright\*, and examine the section containing the Frame interrupt handling routines. You'll notice that if there is no mouse routine installed, the ROM reads the mouse port nine times, before going on to do the keyboard scanning. This ordering is important, as it ensures that you are reading the keyboard when you think you are.

Arguably you should be able to read the mouse, and then the cursor keys, all from the same port. Unfortunately, this is impossible – at least it is with the SAM's current design. Holding down the cursor keys or the Control key will interfere with the mouse reading process. This won't disrupt the mouse data – you just abort the process if you notice that the dummy byte isn't &0F as you'd normally expect it to be.

What it does mean though, is that holding down control while using the mouse won't work, so that's one familiar GUI interface trick that you can't use on the SAM. You could use the EDIT key as a substitute if you wanted, or perhaps one of the SHIFT keys. It's up to you.

\* The ROM source is available from BetaSoft, price £15. See the Contacts section at the end of the mag.

### What if you've not got one?

There is a way to tell if the mouse is plugged in or not, and the way I'm going to do it here is much easier than ways I've done it in the past.

Using a standard mouse driver (like the one above), you could, for instance, read the mouse, (which will clear the offset registers), and then immediately read it again. Check any of the coordinate bytes – if they're not zero, you can assume that there is no mouse plugged in. (This relies, of course, on no keys being pressed, and that you're not moving the mouse really quickly while you're checking – which most people won't be).

The other way to do it is this short routine which I came up with after reverse engineering the mouse interface.

Check for mouse presence:

;Returns Z if mouse present, NZ if not...  
chk4mse:

```
LD BC,&FFFE
LD DE,&0B0F
IN A,(C)
IN A,(C)
AND E
CP E
RET NZ
```

chkmsel:

```
IN A,(C)
DEC D
JR NZ,chkmsel
AND E
RET
```

Next issue: How the mouse hardware actually works.

Simon Cooke.



# Is There Anybody Out There?

Dave Whitmore recently announced the creation of a brand new, SAM orientated bulletin board system – the Dalmation BBS. We decided to get the low-down directly from its source... so here's Dave to tell you what it's all about...

There is a new BBS for SAM users!

Erm... just one snag you might be thinking of... namely that the SAM doesn't have any comms capability yet.

One day soon, though we aren't sure when, software and comms interfaces that allow the use of off the shelf modems may be available. This, along with proposed hard drives will revolutionise everyone's perception of the SAM.

It will also mean that developers and enthusiasts will be able to communicate much faster and generally get things done quicker. For example, we all know how frustrating it can be to wait for letters and disks to arrive in the post...

But for now, if you're already using a modem with your PC, Amiga or ST, then there's good news. The Dalmation Bulletin Board Service

will provide a choice of PD software, message bases, lots of other things too numerous to list mention (including ones we haven't thought of yet). To make things even more interesting, there will be small machine-specific areas too.

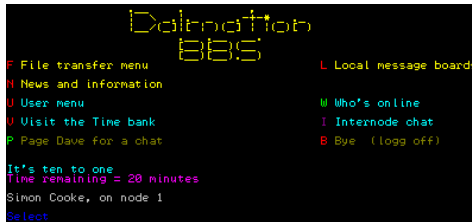
We'll listen to what members want

from the service too. We're open to all ideas and suggestions. Co-sysop status is also available for the serious enthusiast. (Eek... I appear to have been made

co-sysop all of a sudden... I must be a serious enthusiast!!! Cookie)

## What's on-line?

Before I answer that, let me point out that there are NO RATIOS on this BBS. This means that you don't have to upload software in order to download software. Uploads will always be very welcome, and are generally encouraged, but they're not strictly necessary.



Where it all happens on the Dalmation BBS...

Full instructions on how to unarchive and transfer files to SAM disks will be available from the BBS.

To be able to transfer SAM software between machines, you will need to work with PC (720k) disks. There are utilities that will help with the transfer. KE Disk is widely available within the SAM PD scene.

Commercial utilities such as PC\_Suite are also recommended (though not essential). Many BBS's carry large sections of Spectrum Snapshots for use with Amigas and PCs. This will not be one of them, but we may offer SAM 128k Conversions (subject to legality!)

We must stress that this system is in its infancy. To start with this will be a closed system, with local access only. In future we may branch out and link up to networks such as Fidonet, but until there is a sufficient user base, we're going to keep the BBS local only.

### So when's it on-line then?

It sounds bad when anyone says that a BBS is only available on one day a week, but to start off with, that's how

it's going to be. The BBS will have strict opening hours and initially be available on Saturdays, from 12 noon until 12 midnight.

Depending on the success of the BBS, the opening hours may be extended to Sundays, or even the complete 48 hours BT weekend rate period. After that, who knows?

Sessions on the BBS can always be arranged in advance, by voice phoning the Sysop on the same number as the BBS at any other time. Don't forget, you'll get a voice answer outside these hours.

### Are we mad or what?

Okay, all of this might sound eccentric, but come on – most of us make the pilgrimage to Quedgeley Village Hall twice a year! If that's not eccentric, I don't know what is!

I look forward to seeing you on the BBS. The number is: 01744 614150

*Dave Whitmore* (Sysop,  
Dalmation BBS)



```
*****
* The SAM coupe Files menu *
*****

A Utilities
B Games
C Demos
D Miscellaneous

L Link all SAM Areas
P Previous Menu
Q Quit to Main Menu

"It's just after five past one..."
Time remaining: 35 mins Select: A
```

The SAM Coupé files area... lovely eh?

# Subscriptions

## THREE reasons why you should subscribe to *Based On An Idea...*

### It's Cheaper!

To buy each issue separately costs £2.00 x 4 = £8.00

To buy a year's subscription (four issues) costs only **£6.00** – an amazing saving of **£2.00!**

### Less Anxiety!

We send out your copy of the mag as soon as it comes back from the printers – no more wondering “Is the latest issue out yet?” – you'll know because it will pop through your letter box as soon as we can possibly get it to you!

### Guaranteed: No Macaulay Caulkin!

No matter what depths we end up stooping to, we can guarantee that your pristine and shiny fresh-from-the-printers copy of *Based On An Idea* will never have Macaulay Caulkin in it. Except possibly when we really want to have a dig at him. © Recycled YS Jokes Inc.

To subscribe to *Based On An Idea...* all you need to do is send your name and address, with a cheque for £6.00 made payable to M. Rookyard, to the address below. Subscription wrangler extraordinaire Maria Rookyard will take your details and ensure that you receive your very own pristine copy of the latest issue of *Based On An Idea...* with a minimum of fuss and effort on your part, as soon as is possible. Please state which issue you want your subscription to start with in your covering letter.

*Based On An Idea,*

**1 Dovey Close, Astley,  
Tyldesley, Manchester,  
M29 7NP**

# SAM GETS WIRED...

The SAM, believe it or not, does actually have quite a large Internet presence... take the SAM Users Mailing List for example, whose subscribers include Stefan Drissen, Allan Skillman, Ian Collier. Even FRED boss Colin MacDonald makes the occasional appearance...

I'll let you into a little secret – I've even sent emails using my SAM – so it is possible, and I've written the software to do it. But that's not what this article is about. No, the purpose of this is to let you into the big secret – ie how to actually access this mine of information. It's not that difficult really, and seasoned Internet pro's will be wondering why on Earth I'm making such a fuss of it all.

So, assuming that you know how to access all of the facilities which I'm about to mention (and that you have a basic knowledge of how the Internet works\*), let's see what's out there.

## Email

On the email side of things there's only really one place to go – and that's the SAM Users Mailing List. Averaging at least a couple of postings a day, the chat at times has varied from the technical (from ROM bugs, to new Hard-Drive specifications, to whether or not Unix could be done on the SAM), to the silly (why FRED is called FRED, and not, say, FREDSoft), to the sublime... and it's quite easy to access too.

---

\* And if not, why not try getting hold of a copy of Net User issue 5, from Paragon Publishing, which holds the decidedly trimmed version of my Student Guide to the Internet... it's still all valid stuff even if you're not a student!

Plenty of people are on there (at last count about forty or so SAM users all told – though there's only usually about 12 active participants on the list), including Colin Piggot and Brian Gaff, so it's great fun! Kind of like one of those Chat Line thingies, except it's text based, and much, much more interesting...

To sign up, all you need to do is send an email to sam-users-request@nvg.unit.no, with subscribe <your email address> as the actual text of your email. If you really wanted to find out which other SAM users are on the list, you could stick a list command just below your subscribe details.

To actually send any messages, you just mail them off to sam-users@nvg.unit.no. Easy as pie really.

## FTP

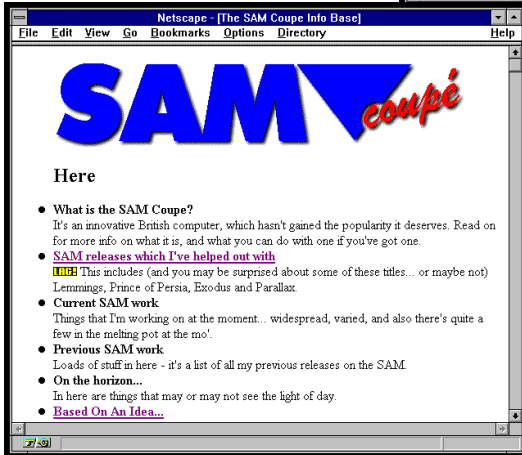
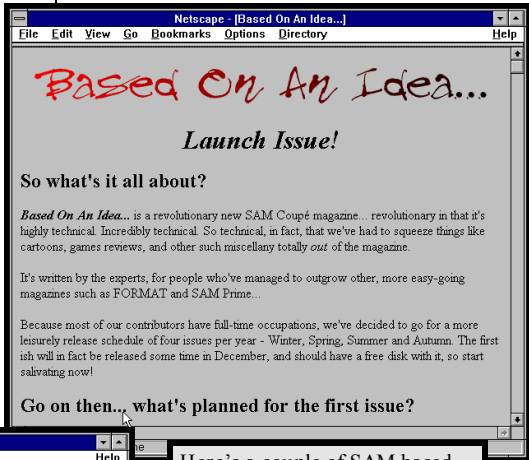
The FTP presence of the SAM is pretty similar to its Email presence – not surprising really, when you consider that the SAM Internet file archive is held on the same machine that handles the mailing list. You can access it by FTPing to ftp.nvg.unit.no, and looking in the /pub/sam-coupe/ directory.

Interesting places to check out include

the demos/Entropy section for some bits and bobs that I've put up there, as well as the incoming directory for all the latest software that people have uploaded.

Please Note: We all know that piracy could kill of the SAM if it ever reached the levels that you can find on other machines. The guys at NVG are pretty strict about it, so don't upload any software where the copyright situation is unclear, or where you'd be breaching copyright – that is, where you'd be pirating the software. This has been

The Web presence of the SAM is steadily increasing, and there's a few Entropy members with home pages



Here's a couple of SAM based web pages... to be precise the Based On An Idea... home page, and the SAM Coupé Info Base, which links off to a few other SAM related sites.

out there (Geoff Winkless for example)... and there's always the Speccy pages to sample and savour if you feel a little nostalgic.

### Telnet

One of my favourite places to loll around on during the day

when I should probably be doing some work is Monochrome BBS. And nope, you don't need a phone-line to access it. This BBS is actually on the Internet, and while it doesn't have a SAM specific area, it does have a quite well-stocked Spectrum one. I'm on there as Spectecj (I used to be on as Spectecjr, but the user-names got truncated, and I never changed mine... I'm just too sentimental, I guess), Tim Paveley (of Sad Snail

a public-service announcement...

### World-Wide Web

The SAM is out there on the World-Wide Web, (to prove it, there's a couple of piccies from our site at the top of this page), and in quite a number of places too... The best thing that I can probably do is to just spoon-feed you a couple of addresses, so there's a box around here with them in...

Productions) is on as  
Unc, and there's some  
other SAM users on there  
too... but for the life of  
me I can't remember who  
they are.

You can find  
Monochrome by  
telnetting to mono.org,  
which will give you a list  
of machines you can log  
in on... pick one and  
telnet to it, then to get to  
the BBS, login as mono  
at the prompt.

### The End?

That's it for the moment,  
and no doubt things will  
hot up in the months to  
come as more and more  
SAM users learn to use  
the comms facilities of  
their machines. Next ish, I'll probably  
be bringing you news of the debates  
from the SAM Users Mailing List, for  
those of you who aren't in the thick of  
it already. And I may even serialise  
some of it on the cover disk, with a  
small reader program, so that you can

## SAM ON THE INTERNET – IN GENERAL...

### Telnet:

You could do much worse than try out the  
Monochrome BBS speccy system on mono.org

### Email:

To subscribe to the SAM Users Mailing List, send  
an email to sam-users-request@nvg.unit.no,  
with the text of your email something like:  
subscribe <your.email.address@place>

### FTP:

The SAM FTP file archive may be found on  
ftp.nvg.unit.no... login as anonymous, with your  
email address as the password. You'll find most  
of the files in the /pub/sam-coupe/ directory –  
and you can upload new material to the site (as  
long as it's not going to cause any copyright  
problems) into the incoming directory.

catch up on what you might have  
missed.

*Simon Cooke* (Email:  
simon.cooke@umist.ac.uk, or on  
Fidonet, I'm Simon Cooke at  
2:250/124.2)

## SAM ON THE WORLD WIDE WEB...

The SAM Coupé Info-Base

<http://jumper.mcc.ac.uk/~simonc/samcoupe>

Based On An Idea...

<http://jumper.mcc.ac.uk/~simonc/boai>

Colin Piggott's Quazar Surround Info Page

[http://www.st-and.ac.uk/~www\\_soc/s/virtual/cgp/quazar/qintro.html](http://www.st-and.ac.uk/~www_soc/s/virtual/cgp/quazar/qintro.html)

Tim Paveley's SAM Coupé Scrap-Book

<http://www.soton.ac.uk/~tsp93ma/Coupe/>

Allan Skillman's Xcoupe page

<http://www.hep.ucl.ac.uk/~ajs/xcoupe/>



# A High Speed SAM?

## Fact or Fiction?

I probably don't have to tell you that today's world has been invaded by the IBM PC and its clones. I must admit that I design all my circuit boards using a 286 laptop, but that's mainly because the SAM does not have any software to cover this field of work.

The other thing the SAM is lagging behind with these days is execution speed - no I'm not criticising it, I'm just comparing. Bruce did an excellent job on the SAM when you consider its cost price against its features, such as graphics resolutions, RAM, midi etc. In fact he actually caused me a headache when building the MultiROM because his designs are so thorough, well thought out and correctly designed.

A lot of the housekeeping functions such as screen display are carried out by the ASIC. Inside the SAM the Z80B processor and the ASIC work together with the processor staying in step with the ASIC by the latter issuing WAIT states to the CPU. As the SAMs video memory is actually part of the general program memory, the ASIC has to commandeer the memory while it grabs video data. The highest priority access is given to the video, otherwise the display would either jump and break up or would have multi-coloured snow specs every time the RAM was accessed. Hence the ASIC can only allow the CPU to access the RAM during screen blanking or when it has read and stored enough bytes to be able to

release the RAM. During this time the display is continued using the stored bytes during one of its video data grabbing bursts; this is known as pipelining.

The result of all this contention is that the SAM runs at a reduced effective execution speed. Even so the SAM works pretty quickly when running most programs.

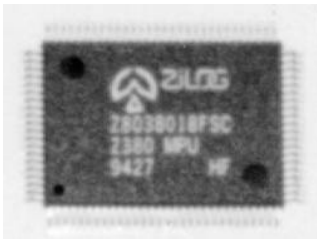
Those of you who were able to attend the FORMAT SAM & Spectrum show in October, would quite probably have seen a SAM with a chunk of Vero board, containing a mass of wires and ICs, plugged in the back. This was the "lets prove an accelerator can work in principle" prototype and it did make most software run much faster. The main problem was that the external Z80C was not always responding to wait states correctly and hence was upsetting the screen display. I'm still working on the problems with the aid of my trusty oscilloscope.

### Which processor?

Okay! Lets assume that we can overcome the problems of using a

high-speed processor, what could we replace it with? The main attributes which the new processor should have are a higher clock speed capability, obviously, and code compatibility. This code compatibility does not necessarily mean another Z80 processor as a high-speed processor from an entirely different family could be used to run a Z80 emulator. Emulators have an unavoidable software overhead (once the software has been written) which reduces its effective operating speed. A Spectrum emulator running on an expensive (£100 approx) 486 processor gives only about a 50% speed increase even though the processor is running 6 times faster. We have been looking seriously at using an ARM710 RISC processor to run an emulator but ...

Hence the only other option is to use a Z80 code compatible high speed



This is the wee beastie. More legs than a caterpillar on steroids...

processor. There are two processors which fill this requirement; the first is the Z80H which runs at up to 20MHz but is virtually unobtainable. The second is the Zilog Z380 which is available at present in an 18MHz version with 25 MHz and 40 MHz versions mentioned in the technical docs which should be available in the

nearish future. The nice part about the Z380 is that it is not much different in price from the Z80H.

## What do we know about this stranger?

The remainder of this article is not about how to get this processor to work with the SAM, that might be covered in a later issue, but it is about the Z380's features and functions; what it will and won't do. I must admit that I have all but fallen in love with the Z380 after having sifted through the technical specs of several processors over the last few months; all this with a view to designing and building Simon Cooke's accelerator board. I shall go through each feature in turn, starting with the main points and working down to the more obtuse.

The first and most important feature is its Z80 code compatibility, which means that any code written on the SAM will work okay with the Z380, provided that the only undocumented op-codes used are the IX and IY register based codes. All the original Z80 op codes work in an identical fashion on the Z380.

If the op codes are compatible then it follows that the register architecture must also be the same. The only exception is that the R register is not incremented on every memory refresh. It is included to allow the LD R,A and LD A,R instructions to work, but the R register now acts as a general purpose 8 bit read/write register.

Memory access - wellll, what can I say other than WOW! The Z380 has a contiguous memory block of a mere 4 Gigabytes, all accessible.

Unfortunately the SAM was designed to fit 512K into a 64K address space by using 32 x 16K pages. The accelerator would be forced to follow this setup, for compatibility, and continue to page 512K into the bottom 64K of the 4 Gigabyte address space. However, all is not lost as the whole address space can still be accessed by using some more specialized new instructions. One of the functions of the ASIC is to multiplex the Z80s 16 bit address bus into a form required by the DRAMs. The Z380 has three outputs which become active during a memory access; these can be used to provide the RAS and CAS timings for the multiplexers.

Another aspect of the memory is that it can be either 8 bits or 16 bits wide. This makes life a lot easier when reading or writing 16 bit values as it reduces the execution times and simplifies the program structure. If, however, you try to do a 16 bit wide read from or write to 8 bit wide RAM, the Z380 will store half the bits then insert an additional read/write cycle for the remaining 8 bits.

The I/O capabilities are very interesting. The processor can slow down its operation during an I/O read or write by a software definable factor; this allows interfaces of different speeds to be used together on the same system. The I/O is split into internal and external ports, the former being used to access special control

registers within the Z380. These registers control functions such as clock prescale, refresh rate, I/O heartbeat rate, register selection, 64K or extended addressing and more. The external I/O address space is 4 Gigabytes with three types of access available. The first two of these are the same as the Z80 using the A or BC registers to provide the addresses and the third works in a similar way but with a full 32 bit address. The data transfers can again be either 8 or 16 bits wide.

The interrupt system is identical to the Z80s as far as the NMI and interrupt modes are concerned. The difference, however, is that the INT input has been renamed INT0 and three new interrupt inputs INT1, INT2 and INT3 have been added. Only INT0 is capable of maintaining the three interrupt modes selected by instructions IM0, IM1 and IM2; the other three operate using vectors - more details later.

The register sets have been extended substantially. The BC, DE, HL, IX and IY have been doubled in length to 32 bits each; the extra 16 bits are referred to as BCz, DEz, HLz, IXz and Iyz respectively. The Z380 has four complete sets of registers, including the alternate registers, of which one set can be accessed at a time. There are also four sets of A, A', F and F' registers, one for each bank of main registers. The Accumulator is still only 8 bits wide but 16 and 32 bit accumulator functions can be taken care of by using the HL and HLz registers. These features not only

speed programs up by virtue of allowing double width arithmetic operations but also a frequently called routine can be assigned a set of registers which it uses exclusively. This does away with the need to PUSH and POP the registers to the stack on entry or exit to the routine, as you simply switch to a different set of registers. The IX and IY registers are slightly different to the other registers in that they are selected independently of the main register banks and of each other.

The I register has also been extended by 16 bits to 24 bits to allow the interrupt vector to operate over the full address range. The Program Counter PC is 32 bits wide but this can be restricted by using the Z380 in what is called native mode. This effectively restricts the PC to 16 bits (ie 64K ) and address 0000FFFF hex overflows to 00000000 (as with the Z80) as opposed to 00010000 hex. The Stack pointer is also 32 bits wide; the extension to SP is called, surprisingly, SPz just like the main registers and PC is extended by register PCz.

The Z380 has a range of internal control registers which allow the software to specify various parameters. Most of these are used to setup various hardware configurations such as RAM/ROM access times and refresh rate, I/O peripheral clock speed and other miscellaneous functions.

The refresh system for DRAM no longer uses the R register but relies on

the internal refresh counters built into most modern DRAMs. The refresh request rate can be determined as between 4 and 4096 bus clock periods in steps of four (ie  $n \times 4$ ) while each refresh request consists of between 1 and 64 refresh transactions. If any refresh requests have been missed ie due to a bus request then the number missed is counted (up to 255) and when bus control is restored, the missing refreshes are generated. The refresh uses the CAS before RAS method as the Z380 is capable of generating these signals direct with very little external logic.

I have tried to keep this to the main features and not get too involved in the intricacies of the processor. Zilog have described the processor far better than I could and have filled two entire manuals in doing so. If an accelerator board is built using the Z380 then we will run a series of both hardware and software articles giving the real nitty-gritty stuff.

Now the most important feature about the Z380 is that they do actually exist. I have found a supplier and have two Z380s in a small antistatic box in my workshop. The only problem is that they are surface mount technology in the form of a 100 pin QFP (Quad Flat Pack) package and through hole sockets are expensive.

Next issue I shall give an outline of how the SAM Accelerator will work. Till then, I wish you an empty bit bucket!

*Martin Rookyard*

# Contacts!

All postal addresses are in the UK unless otherwise noted, as are all phone numbers. If you're dialling from outside the UK, dial your international dialling code, followed by 44, then the UK number (minus the leading zero). And don't forget to send an SAE with your correspondence if you want a reply.

## Software

Atomik Software, ☐ Dept.  
BOAI, 20 Grove Road,  
Hoylake, Wirral,  
Merseyside, L47 2DT  
B.G. Services, ☐ 64  
Roebuck Road,  
Chessington, Surrey, KT9  
1JX (please write all letters  
/ orders clearly) ☎ (0181)  
2874180  
Elysium Software (SAM  
Adventures), ☐ 50  
Chadswell Hgts, Lichfield,  
Staffs, WS13 6BH  
F9 Software, ☐ 18 Mill  
Lane, Glenburn Rd,  
Skelmersdale, Lancs, WN8  
8RH ☎ (01695) 31163  
FRED Publishing, ☐ 40  
Roundyhill, Monifieth,  
Dundee, DD5 4RZ ☎  
(01382) 535963  
Hilton Computer Services  
Ltd, ☐ 3 Suffolk Drive,  
Guildford, Surrey, GU4  
7FD ☎ (01483) 578983  
Jupiter Software, ☐ 2  
Oswald Rd, Rushden,  
Northants, NN10 0LE  
Persona, ☐ 31 Ashwood  
Drive, Brandleshome, Bury,  
Lancs, BL8 1HF ☎ (0161)  
797 0651  
Revelation Software, ☐ 45  
Buddle Lane, Exeter, EX4  
1JS  
Saturn Software, ☐ 5  
Ivanhoe Drive, Westfields,  
Ashby de la Zouch, Leics.,  
LE65 2LT

S.D. Software, ☐ 70  
Rainhall Road,  
Barnoldswick, Lancashire,  
BB8 6AB  
Steve's Software, ☐ 7  
Narrow Close, Histon,  
Cambridge, CB4 4XX ☎  
(01223) 235150 from 6pm-  
9pm, Mon-Sat  
Supplement Software, ☐ 37  
Parker St, Bloxwich,  
Walsall, WS3 2LE ☎  
(01922) 406 239

## Hardware

B.G. Services (see under  
Software)  
Colin Piggot, ☐ 204  
Lamond Drive, St.  
Andrews, Fife, KY16 8RR  
Rooksoft, ☐ 1 Dovey  
Close, Astley, Tyldesley,  
Manchester, M29 7NP ☎  
(01942) 886084  
West Coast Computers, ☐  
see FORMAT Publications.

## Publications

Adventure Probe, ☐  
Barbara Gibb (Editor),  
Adventure Probe, 52  
Burford Road, Liverpool,  
L16 6AQ  
Based On An Idea... ☐ 1  
Dovey Close, Astley,  
Tyldesley, Manchester,  
M29 7NP (or email us at  
entropy@jumper.mcc.ac.uk  
) ☎ (01942) 886084  
Crashed, ☐ 16 The  
Avenue, Manston, Leeds,

LS15 8JN ☎ (0113) 232  
6726  
FRED Disk Magazine, ☐  
See FRED Publishing.  
FORMAT Publications, ☐  
34 Bourton Road,  
Gloucester, GL4 0LE ☎  
(01452) 412572 ☎ (01452)  
380890  
Zodiac, ☐ New House,  
Holbear, Chard, Somerset,  
TA20 2HS ☎ (01460)  
62118

## Books & Manuals

B.G. Services (see under  
Software) for genuine  
Digital Research CP/M 2.2  
manuals  
FORMAT Publications (see  
under software) for SCADS  
manuals

## Demo Groups

Entropy, ☐ 18 Braemar  
Drive, Sale, Cheshire, M33  
4NJ (email:  
entropy@jumper.mcc.ac.uk  
)  
MNEMOtech, ☐  
MNEMOtech, c/o Andrew  
Collier, 57 Wyndham  
Avenue, Bolton, BL3 4LG.  
☎ (01204) 652470

☐ – Postal address (or  
email address) • ☎ –  
Telephone number • ☎ –  
Fax number

# Next Issue

- **The SAM Mouse – next issue we explain the hardware inside the interface box.**
- **Instruction Timings – the definitive guide**
- **An update on the Dalmation BBS**
- **The SAM Accelerator – the principles behind it.**
- **Reviews, News, Letters, our very own help page and much much more!\***

**Available from the end of April  
£2.00 per issue, or see our  
subscription offers!**

**\*Of course, that is, assuming that people write in with letters and queries for the help page... Hint! Hint!**

---

## Index of Advertisers

IFC Rooksoft, 1 Dovey Close, Astley, Tyldesley, Manchester, M29 7NP, UK  
6 SAM PD, 18 Mill Lane, Glenburn Rd, Skelmersdale, Lancs, WN8 8RH, UK  
21 Stefan Drissen, Zevende Herven 6, 5232 JZ 's-Hertogenbosch, The Netherlands  
IBC Persona, 31 Ashwood Drive, Brantleshome, Bury, Lancs, BL8 1HF, UK  
BC FRED Publishing, 40 Roundyhill, Monifieth, Dundee, DD5 4RZ, UK

IFC= Inside Front Cover, IBC = Inside Back Cover, BC = Back Cover

---

Editors: Simon Cooke, Martin Rookyard • DTP: Simon Cooke • Contributors: Simon Cooke, Stefan Drissen, Martin Rookyard, Maria Rookyard, Dave Whitmore • Text proof reading: Maria Rookyard • Technical proof reading: Simon Cooke, Martin Rookyard • Tel: (01942) 886084 • Address: 1 Dovey Close, Astley, Tyldesley, Manchester, M29 7NP • Email: [entropy@jumper.mcc.ac.uk](mailto:entropy@jumper.mcc.ac.uk) • WWW: <http://jumper.mcc.ac.uk/~simonc/boai/> • Subscriptions to: Maria Rookyard at the above address – cheques made payable to M. Rookyard please! • Special Offer: Take out a year's subscription – that's 4 issues of the magazine – for only £6.00! • Published by Lame Ducks International Press • Printed by Manchester Computing Centre • Based On An Idea... is a quarterly magazine from those talented bods at Entropy and Rooksoft.

©1996 Entropy, Rooksoft • ©1996 Lame Ducks International Press

---

# BLITZ<sup>80</sup>

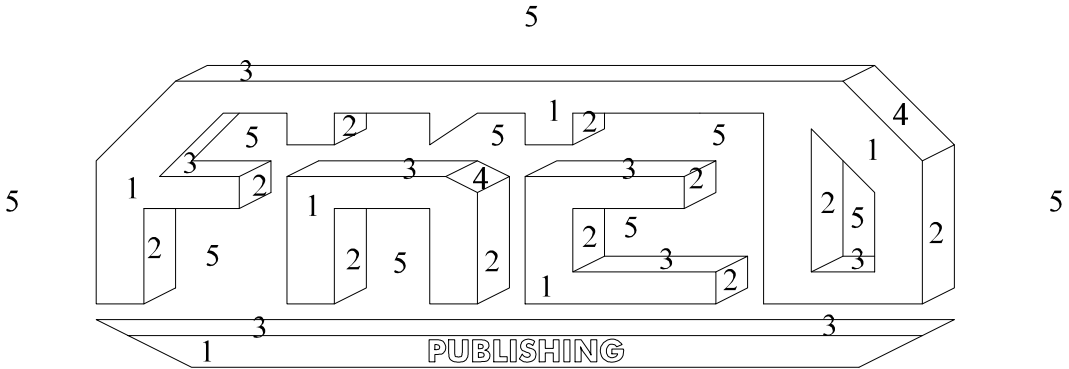
The New  
Bi-Monthly Disk  
Magazine  
from

# PERSONA

Persona, 31 Ashwood Drive, Brandleshome, Bury,  
Lancs, BL8 1HF. Tel: (0161) 797 0651.

Formerly SAM Prime, Blitz magazine is available for £2.00 per issue from Persona.  
Cheques made payable to M. D. MacKenzie. We could tell you what was in it, but then  
we'd have to kill you.

# Painting By Numbers



**FRED Publishing presents two utilities for  
SAM owners of the artistic persuasion**

## **SAM PAINT**

**The astounding paint package which beats seven shades (or even 16 colours) out of 16 bit packages! Includes perspective, plasma and gradient fill and image deformation options with which you can create an amazing array of graphical effects.**

**£25.00**

## **SAM SPRITE**

**Create sprites for use with SCADS, GamesMaster and other game creation utilities! Includes comprehensive animation and screen grabbing features, mouse control and more!**

**£10.00**

FRED Publishing, 40 Roundyhill, Monifieth, Dundee, DD5 4RZ  
Tel: 01382 535963 (Evenings and weekends only) Fax: 01382 535622  
Cheques made payable to FRED Publishing